



SAVONIA

PUHTI

Android-sovellus

TEKIJÄ: Niko Snellman

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä Niko Snellman	
Työn nimi Puhti, Android- sovellus	
Päiväys 6.5.2018	Sivumäärä/Liitteet 29
Ohjaaja(t) Jussi Koistinen, Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) Physiobit Oy	
Tiivistelmä	
<p>Opinnäytetyön aiheena oli luoda Android- sovellus konsultointi käyttöön. Sovelluksen päätarkoituksena on ladata opiskelumateriaalia ja tehtäviä laitteeseen. Sovellusta testattiin materiaalilla, joka on tarkoitettu enemmän liikunnalliseen tarkoitukseen. Esimerkiksi käyttäjä voi ladata opiskelumateriaalia liittyen johonkin tiettyyn ruumiin osaan ja suorittaa tehtäviä liittyen materiaaliin. Tehtävät suoritetaan tehtävässä annetuin ohjeiden perusteella. Tämän sovellus on tarkoitettu konsulttien asiakkaille. Tämän sovelluksen nimenä on Puhti.</p> <p>Sovelluksen rakentamiseen käytettiin Android Studio ohjelmaa. Sovellus ohjelmoitiin Java ohjelmointi kielellä. Rajapintojen hallintaan käytettiin Microsoft Azuren palveluita. Rajapinnat ohjelmoitiin käyttäen Node.js koodikirjastoja.</p> <p>Tämä opinnäytetyö on osa suurempaa kokonaisuutta, johon kuuluu valmentajan sovellus. Valmentajan sovelluksella luodaan kaikki materiaalit ja tehtävät mitkä näkyvät myöhemmin asiakkaan sovelluksessa. Samaan aikaan valmentajan sovellusta teki toinen työntekijä. Yhteistyötä piti tehdä paljon varsinkin rajapintojen suunnittelussa, että saataisiin yhteensopivuus molempien sovelluksien välillä. Opinnäytetyöstä tuli toimiva sovellus ja kaikki vaaditut ominaisuudet löytyivät sovelluksesta. Sovelluksen tekemiseen meni kauemmin mitä oli suunniteltu.</p>	
Avainsanat Puhti, Android, Java, Azure, XML, JavaScript	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author Niko Snellman			
Title of Thesis Puhti, Android application			
Date	6 May 2018	Pages/Appendices	29
Supervisor(s) Mr Jussi Koistinen, Senior Lecturer and Mr Keijo Kuosmanen, Senior Lecturer			
Client Organisation /Partners Physiobit Oy			
<p>Abstract</p> <p>The purpose of the thesis was to make an Android application for consultants. Users can download the studying materials and assignments to the device. This application is intended for sporting purposes. For example, users can download studying materials related to a certain muscle area and perform tasks related to the material. Assignments will be performed according to the instructions that are given in the assignments. This application is for customers of consultants. The name of the application is Puhti.</p> <p>The application was made with Android Studio using Java as the programming language. Microsoft Azure was used to manage the back end of the project. The back end was programmed using Node.js libraries.</p> <p>This thesis is part of a bigger project which includes an application for the trainer. All the materials and assignments are created in the trainer's application and they can be later downloaded in the customer's application. The trainer's application was made by another author. It was necessary to co-operate in planning and programming the back end to get compatibility with both applications. In the end the application was completed with all required features. It took longer than originally planned to make this application.</p>			
Keywords Puhti, Android, Java, Azure, XML, JavaScript			

ESIPUHE

Isot kiitokset Physiobit Oy:lle, joka tarjosi mahdollisuuden tehdä suuren projektin Android- laitteelle. Projektin aikana sain paljon kokemusta Android- sovelluskehityksestä ja myös rajapintojen toiminnasta. Tämän opinnäytetyön laajuus oli aika suuri verrattuna mitä sen olisi pitänyt olla, mutta sain tehtyä tämän valmiiksi ajoissa.

Kuopiossa 3.5.2018

Niko Snellman

SISÄLTÖ

1	JOHDANTO	6
2	TYÖKALUT	7
2.1	Android Studio	7
2.1.1	Käyttöliittymän luominen	7
2.1.2	Android- emulaattori	8
2.1.3	Gradle- työkalu	9
2.1.4	SQLite- tietokanta	9
2.2	Microsoft Azure	9
2.2.1	Rajapinta.....	10
2.2.2	Tietokanta	11
2.2.3	Azure Storage.....	11
3	PUHTI	13
3.1	Kirjautuminen ja rekisteröinti	13
3.2	Sisällön lataaminen	15
3.3	Etusivu	16
3.4	Ryhmät	17
3.5	Materiaalit	19
3.6	Tehtävät	21
3.7	Viestit	24
3.8	Asetukset	26
4	YHTEENVETO.....	28
	LÄHTEET JA VIITTAUKSET	29

1 JOHDANTO

Tämän raporin alussa käydään läpi työkaluja ja tekniikoita mitä käytin tämän opinnäytetyön tekemisessä. Työkaluista käydään läpi vai ne ominaisuudet ja palvelut, mitä käytin tässä opinnäytetyön tekemisessä. Tämän jälkeen käydään läpi itse sovellusta ja sen toiminnallisuutta. Tässä raportissa myös selitetään mitä tapahtuu palvelimen puolella, kun sovellus lähettää dataa palvelimelle.

Opinnäytetyön aiheena oli luoda koulutus sovellus Androidille. Sovelluksen avulla käyttäjä voi tarkastella hänelle tarkoitettua materiaalia ja suorittaa materiaaliin liitettyjä tehtäviä. Sovelluksen nimenä on Puhti ja koko Puhti- projektiin liittyy valmentajan sovellus ja asiakkaan sovellus. Valmentajan sovellus on tarkoitettu web- selaimelle ja asiakkaan sovellus tehdään Androidille, mikä liittyy kyseiseen opinnäytetyöhön. Opinnäytetyöhön liittyi sovelluksen lisäksi palvelimen rajapinnan suunnittelu ja ohjelmointi.

Opinnäytetyön tarjosi Physiobit Oy niminen yritys. Physiobit Oy luo sovelluksia kaikille alustoille. Yritys on erikoistunut luomaan sovelluksia liittyen liikuntaan ja hyvinvointiin.

2 TYÖKALUT

2.1 Android Studio

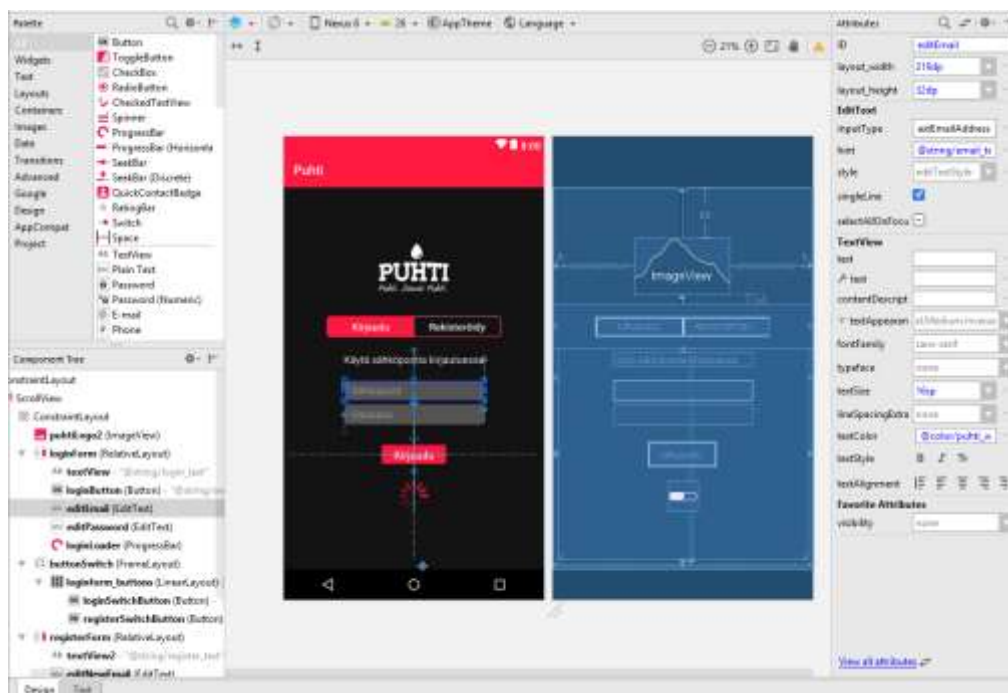
Android Studio on sovelluskehitystyökalu, jolla voi kehittää sovelluksia Android- käyttöjärjestelmälle. Sen voi asentaa Windows, Linux tai Mac laitteeseen. Android Studiosta löytyy Android SDK (Software Development Kit) työkalut, jotka on tarkoitettu Android sovelluksien koodaamiseen. Koodikielenä käytetään Java- tai Kotlin ohjelmointikieltä sovelluksien rakentamiseen. Tässä opinnäytetyössä käytettiin Android Studiota sovelluksen tekemiseen. (Android Developers, ei pvm.)

Käytin seuraavia Android Studion tekniikoita opinnäytetyön tekemiseen:

- Graafinen käyttöliittymän rakentaja
- Android- emulaattori
- Gradle- työkalu
- SQLite- kanta

2.1.1 Käyttöliittymän luominen

Android Studiolla voi rakentaa käyttöliittymiä, joko graafisella käyttöliittymänrakentajalla tai XML- kuvauskielellä. Graafisella työkalulla voi vetää komponentteja näyttökuviin ja asettaa niille skaalautuvuus sääntöjä, jotta komponentit skaalautuvat hyvin eri kokoisissa näytöissä. Käyttöliittymän rakentajalla ei voi luoda toiminnallisuutta komponentteihin. Toiminnallisuus koodataan Java- kielellä Java- tiedostojen kautta. Käyttöliittymään voi asettaa erilaisia teemoja, jotka määräävät eri komponenttien värisävyt. Kaikilla käyttöliittymän komponenteilla on omat ominaisuudet, joita voi käydä säätämässä haluamansa mukaan. (Android Developers 2018, 4.)



KUVA 1. Android Studio näytön muokkaus näkymä

Toinen vaihtoehto käyttöliittymän rakentamiseen on luoda ne XML- kuvaus kielellä. Komponentteja voi lisätä näyttöön kirjoittamalla XML- koodia tiedostoon. Komponenttien ominaisuuksia säätää suoraan koodista. Joskus joitakin ominaisuuksia on vaikea säätää graafisen työkalun avulla, joten XML- tiedoston avulla on helppo säätää komponentin ominaisuutta muokkaamalla koodia.

```

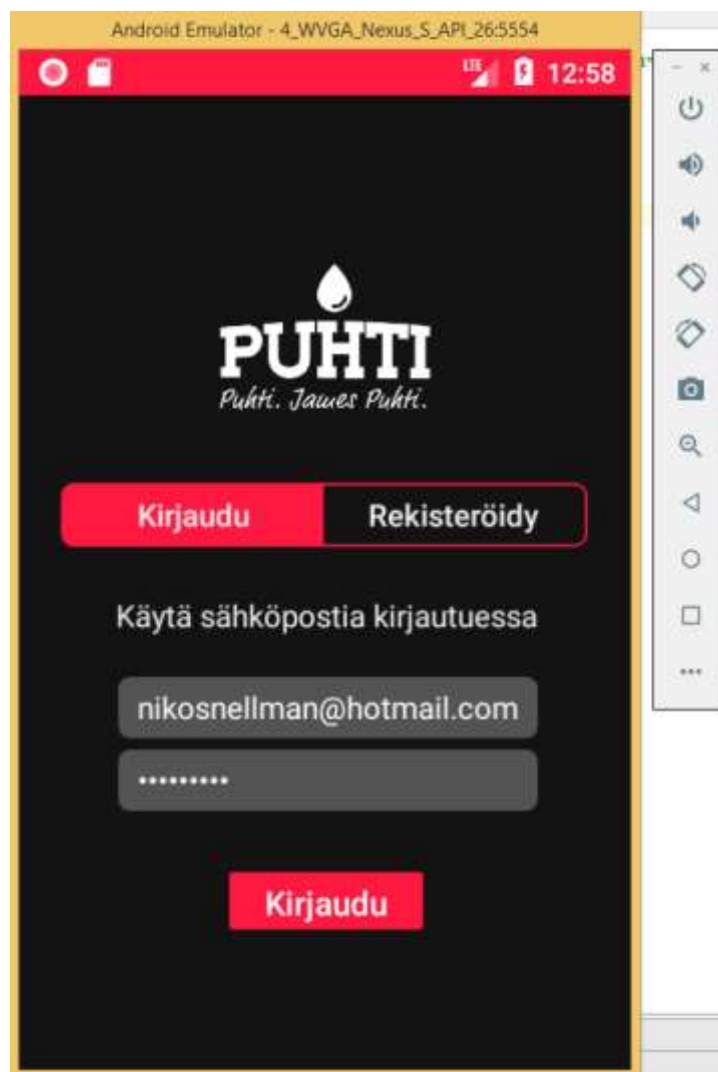
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="@color/puhti_dark_grey"
8      android:visibility="visible"
9      tools:context="fi.physibit.puhti.Login">
10
11      <ScrollView
12          android:layout_width="5dp"
13          android:layout_height="5dp"
14          app:layout_constraintBottom_toBottomOf="parent"
15          app:layout_constraintEnd_toEndOf="parent"
16          app:layout_constraintStart_toStartOf="parent"
17          app:layout_constraintTop_toTopOf="parent">
18
19          <android.support.constraint.ConstraintLayout
20              android:layout_width="match_parent"
21              android:layout_height="match_parent">
22
23              <ImageView
24                  android:id="@+id/puhtiLogo"
25                  android:layout_width="151.5dp"
26                  android:layout_height="88.5dp"
27                  android:layout_marginEnd="8dp"
28                  android:layout_marginStart="8dp"
29                  android:layout_marginTop="88dp"
30                  android:contentDescription="Puhti"
31                  app:layout_constraintEnd_toEndOf="parent"
32                  app:layout_constraintStart_toStartOf="parent"
33                  app:layout_constraintTop_toTopOf="parent"
34                  app:srcCompat="@drawable/puhti_logo_transparent" />
35
36              <RelativeLayout
37                  android:id="@+id/loginForm"
38                  android:layout_width="395dp"
39                  android:layout_height="333dp"
40                  android:layout_marginBottom="8dp"
41                  android:layout_marginEnd="8dp"
42                  android:layout_marginStart="8dp"

```

KUVA 2. XML- tiedosto

2.1.2 Android- emulaattori

Android- emulaattori on virtuaalinen versio Android laitteesta. Android Studioon voi asentaa eri versioita Android- laitteista ja testata omaa sovellusta Androidin emulaattorissa. Toinen tapa testata sovellusta on asentaa se fyysiseen laitteeseen Android Studioon kautta. Androidin emulaattorilla voi tehdä suurimman osan asioista mitä oikealla Android- laitteella voi tehdä. Emulaattorissa kuitenkin ei ole ominaisuuksia mitkä vaativat fyysisen laitteen. Ominaisuudet mitä emulaattorissa ei ole ovat Bluetooth, NFC, SD- kortin lisäys, kuulokkeiden lisäys ja USB. Emulaattorin oikealla puolella näkyy nappeja, joilla voi esimerkiksi säätää äänenvoimakkuutta tai kääntää laitteen vaakatasoon. (Android Developers 2018, 5.)



KUVA 3. Android Emulaattori

2.1.3 Gradle- työkalu

Gradle- työkalu on automaattinen projektin rakennus työkalu. Gradle- työkalun kautta voi lisätä projektiin useita eri koodikirjastoja tai uudelleen käyttää vanhaa koodia pienellä vaivalla. Gradle toimii itsenäisesti ilman Android Studiota. Sen avulla voi rakentaa omia Android- sovelluksia vaikka komentokehotteen avulla ilman Android Studiota. (Android Developers 2018, 4.)

2.1.4 SQLite- tietokanta

SQLite- tietokantaa käytettiin opinnäytetyössä tietojen tallentamiseen paikallisesti laitteeseen. SQLite- kanta ei ole Androidin oma ominaisuus vain sitä voi käyttää myös eri alustoilla. Sitä voi käyttää julkisissa tai yksityisissä projekteissa.

2.2 Microsoft Azure

Microsoft Azure on alusta, joka tarjoaa monipuolisia pilvipalveluita. Sillä pystyy luomaan, julkaisemaan ja hallitsemaan sovelluksia palvelimien kautta. Oman palvelun toimintaa pystyy seuraamaan Azure Monitor ominaisuuden kautta. Esimerkiksi sieltä näkee kaikki rajapinta kutsut ja sisään ja ulos menevä data. (Microsoft Azure 2018.)

Käytin seuraavia Azuren palveluita opinnäytetyön tekemiseen:

- Azuren rajapinnat
- Tietokanta
- Azure Storage

2.2.1 Azuren rajapinnat

Microsoft Azuressa on rajapinnoille tarkoitettu palvelu nimeltään Web Apps. Sen kautta voi luoda REST palveluita ja tehdä palvelin pyyntöjä oman sovelluksen kautta palvelimelle. Web Apps tukee monta eri koodikieltä. Tässä opinnäytetyössä käytin rajapintojen koodaamiseen Node.js koodikirjastoja. Rajapintojen koodit voi myös tallentaa versiohallintaan, jotta tiedot pysyvät aina tallessa. Rajapinnoilla on myös mahdollisuus säätää oikeuksia, että ketkä saa käyttää tiettyä rajapintaa. Kuvassa 4 näkyy mahdollisuudet, mitä oikeuksia eri tyyppisille REST pyynnöille voi antaa. Rajapinnasta voi tehdä avoimen eli kaikki käyttäjät voi päästä käsiksi rajapintaan tai käyttöä voi rajoittaa vain kirjautuneille käyttäjille. Rajapinnat käyttävät Azuren omaa todennusta kirjautumisen tunnistuksessa. (Microsoft Azure 2017, 1.)

★ Name

GET permission

Allow anonymous access

POST permission

Allow anonymous access

Allow anonymous access

Authenticated access only

Disabled

PATCH permission

Allow anonymous access

DELETE permission

Allow anonymous access

KUVA 4. Rajapinnan käyttöoikeudet

2.2.2 Tietokanta

Azuren kautta pystyy lisäämään ja muokkaamaan tauluja tietokantaan. Tietokannan dataan pääsee käsiksi rajapintojen kautta tai suoraan palvelimelta. Rajapintojen avulla voi suorittaa SQL-komentoja, joilla voi viedä tai tuoda dataa rajapintaan. Tietokanta pyörii erillisellä palvelimella, jota pystyy hallinnoimaan Azuren IaaS palvelun kautta.

2.2.3 Azure Storage

Azure Storage on tarkoitettu suuren datan tallentamiseen palvelimelle. Azure Storagen kautta voi tallentaa esimerkiksi kuva tiedostoja tai dokumentteja. Kaikki tiedostot ladataan tai tallennetaan https yhteydellä. Tiedostot tallennetaan omaan containeriin. Container vastaa kansiota, jossa on kaikki tiettyyn kansioon kuuluvat tiedostot tallessa. (Microsoft Azure 2018, 4.)

Azure Storageen tallennetut tiedot ovat salattuja ja niihin pääsee käsiksi SAS (*Shared Access Signature*) osoitteen avulla. Kuvassa 5 näkyy esimerkki URI osoite, jossa on SAS mukana. Osoitteen ensimmäisessä osassa on polku tiedoston resurssiin. Resurssin polussa lukee palvelun domain osoite, container ja ladattavan tiedoston nimi. URI osoitteen kysymysmerkin jälkeen tulee SAS-osoite, jonka avulla saadaan lupa ladata tiedosto palvelimelta. Osoitteen loppuosassa näkyy päivämäärät, jotka kertovat luomispäivämäärän ja vanhentumispäivämäärän. Kun aika umpeutuu niin osoitteesta tulee käyttökelvoton. Osoitteessa näkyy myös, että mitä oikeuksia osoitteella on. Kyseisessä osoitteessa lukee rwd eli osoitteella on luku, kirjoitus ja poisto oikeudet. Osoitteen

lopussa oleva merkkijono on allekirjoitus, jonka avulla annetaan käyttäjälle oikeus ladata tiedosto.
(Microsoft Azure 2017, 4.)

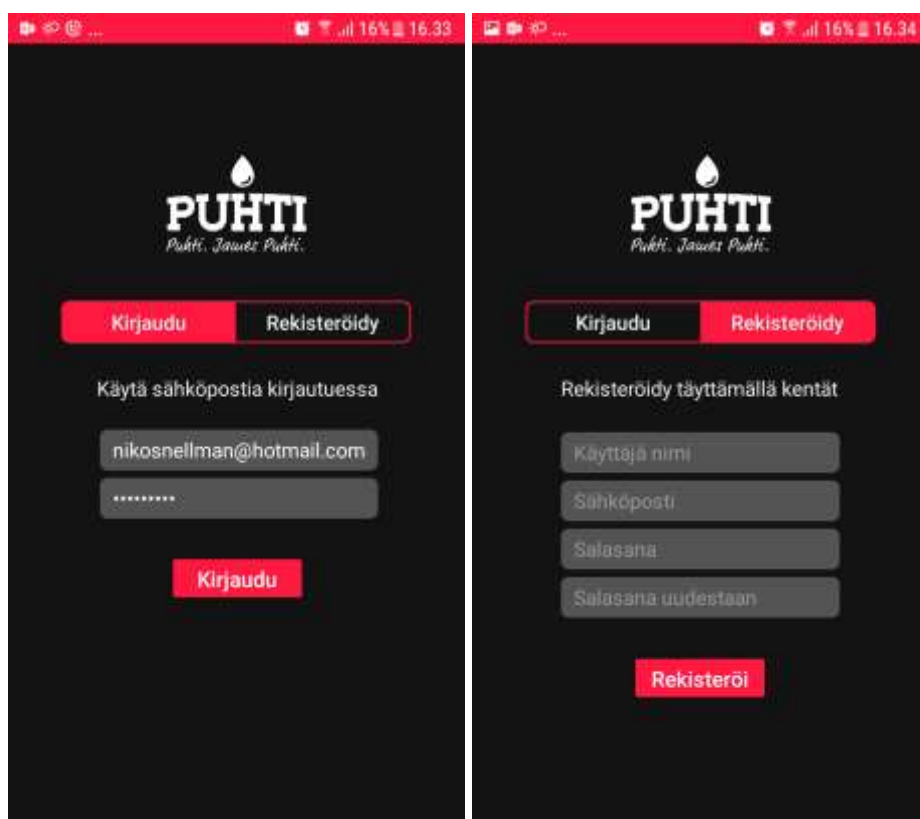
<https://domain/container/tiedosto.pdf?st=2018-02-28T12%253A19%253A01Z&se=2018-02-28T12%253A34%253A01Z&sp=rwd&sv=2016-05-31&sr=c&sig=eyYWfywfvvfiVfyiFWFvYFIUfiywsVfiyF>

KUVA 5. Shared Access Signature

3 PUHTI

3.1 Kirjautuminen ja rekisteröinti

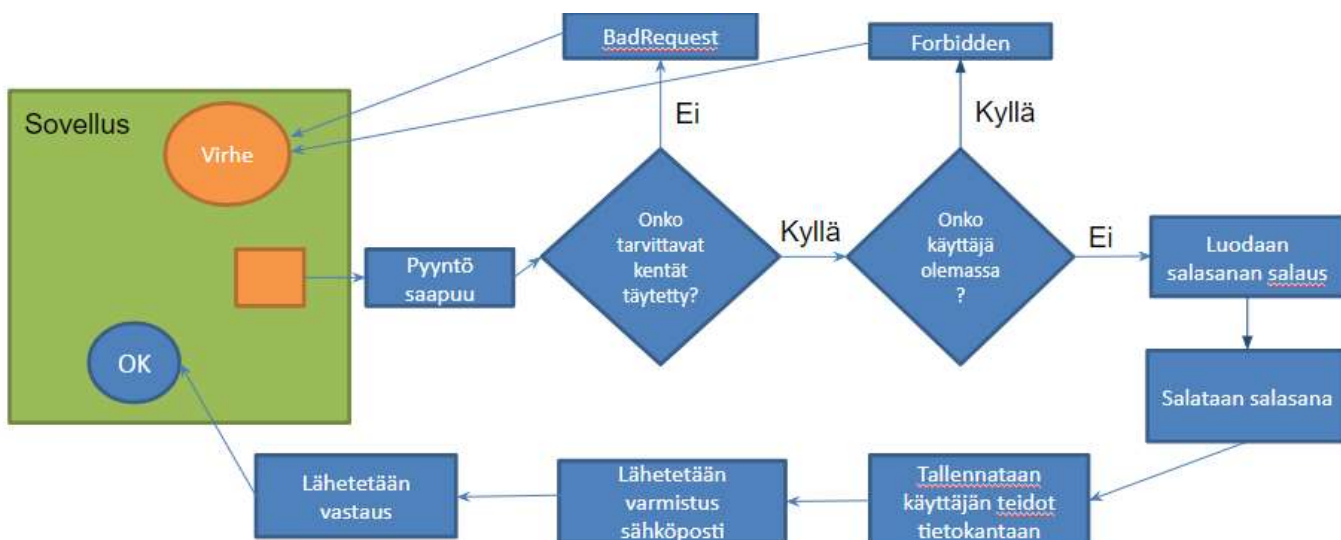
Sovelluksen ensimmäisenä näyttönä on kirjautumis- ja rekisteröitymisnäyttö. Tällä sivulla voi joko kirjautua tai rekisteröidä uuden käyttäjän. Sovellusta ei voi käyttää ilman kirjautumista. Kuvassa 6 Puhti logon alapuolella on 2 nappia, joilla voi vaihtaa kirjautumis- tai rekisteröintinäkömään päälle. Sovellus piilottaa aina toisen näkömään näkymättömiin ja näyttää toisen näkömään käyttäjälle.



KUVA 6. Kirjautumis- ja rekisteröintinäytöt

Rekisteröinnissä vaaditaan käyttäjänimi, sähköposti ja salasana 2 kertaa. Kaikki kentät ovat pakollisia. Jos jotain jää täyttämättä niin sovellus ilmoittaa, että tietoja puuttuu, jos yrittää painaa rekisteröinti nappia. Jos sähköposti on kirjoitettu väärin tai salasanat eivät täsmää niin silloin tulee myös virhe ilmoitus. Kun kaikki kentät on täytetty oikein niin rekisteröinti napista sovellus lähettää palvelimelle pyynnön uudesta rekisteröitymisestä.

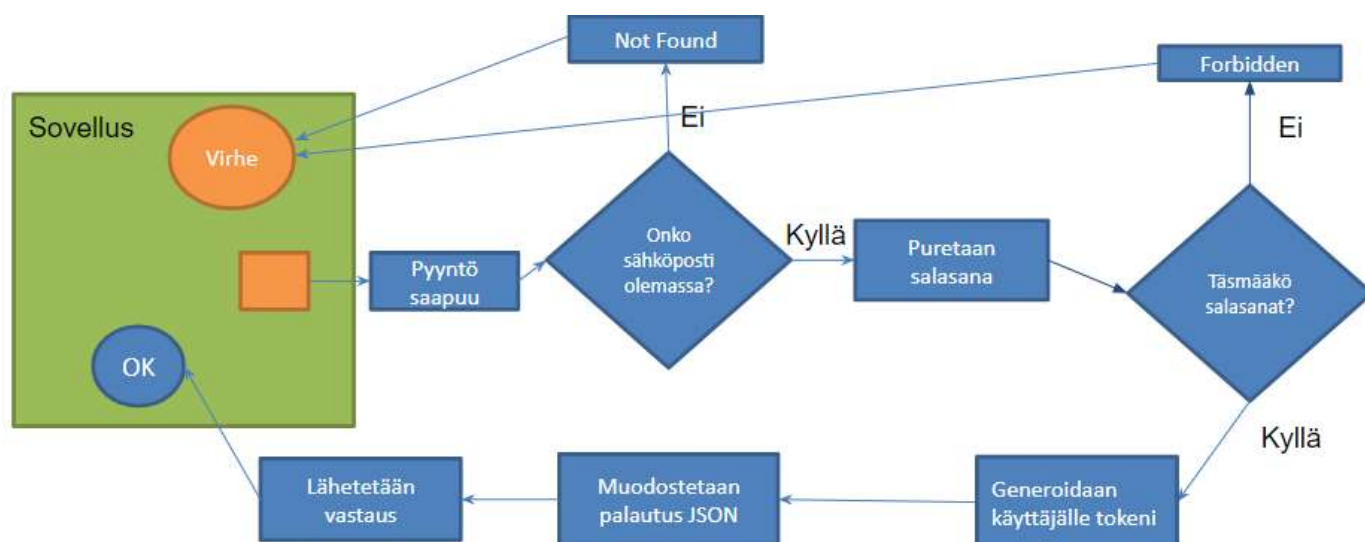
Ennen kuin lähetetään palvelimelle mitään kutsua niin lähetettävä data pitää muodostaa JSON-muotoon. Sovelluksen kaikki palvelin pyynnöt on tehty REST- palvelulla. Rekisteröinnin pyyntö lähetetään POST tyyppisellä HTTP- pyynnöllä. Rekisteröinti tietojen data menee POST- pyynnön runkoon. Toisin kuin URL osoitteeseen sijoitettavien parametrien kohdalla, HTTPS-protokolla salaa rungon sisällön, jolloin tieto on luettavissa vasta palvelimella.



KUVA 7. Rekisteröinti vuokaavio

Kuvassa 7 näkyy, että mitä tapahtuu palvelimen päässä, kun sovelluksesta tulee rekisteröitymispyyntö. Palvelimen päässä ensimmäisenä tarkistetaan, että onko tarvittavat tiedot lähetetty sovelluksesta. Jos jotain pakollisia kirjautumistietoja puuttuu, niin palvelin lähettää sovellukseen "BadRequest" vastauksen eli pyyntö on puutteellinen. palvelin tämän jälkeen tarkistaa, että onko käyttäjän sähköpostiosoite jo olemassa. Jos sähköposti jo löytyy niin palvelin lähettää "Forbidden" vastauksen eli pyyntö ei ole sallittu. Jos kaikki tiedot ovat oikean laiset niin aletaan salaamaan käyttäjän antama salasana. Ennen kuin salataan salasana niin pitää luoda salasanan salaus. Tällä tarkoitetaan satunnaisesti generoitua merkkijonoa. Tämän avulla salasana muunnetaan erilaiseen muotoon, että kukaan ei tämän jälkeen suoraan näe toisen salasanaa edes tietokannasta. Kun salasana on salattu niin palvelin tallentaa rekisteröinti tiedot tietokantaan. Tietokantaan tallennetaan myös salattu salasana ja salasanan salaus, jolla saadaan salasana purettua kirjautumisen yhteydessä. Käyttäjän antama salasana pysyy käyttäjän omassa tiedossa. Lopuksi palvelin lähettää käyttäjän antamaan sähköpostiin viestin, jossa on vahvistuslinkki. Linkkiä painamalla rekisteröity käyttäjä tulee käyttöön.

Kun käyttäjän tunnukset on otettu käyttöön, niin sovelluksella voi kirjautua sisään. Kirjautumiseen käyttäjältä vaaditaan sähköpostiosoite ja salasana, jotka syötetään kuvassa 6 näkyviin kenttiin. Kun käyttäjä painaa kirjautu nappia niin sovellus lähettää pyynnön palvelimelle kirjautumisesta. Kuvassa 8 näkyy kirjautumisen vuokaavio. Ensimmäisessä vaiheessa tarkastetaan, että onko annettua sähköpostia olemassa. Jos sähköpostia ei löydy niin palvelin lähettää sovellukseen virhe viestin ja sovelluksessa ilmoitetaan, että sähköposti tai salasana on väärä. Seuraavassa vaiheessa puretaan sähköpostiin liittyvä salasana Hash muotoon ja verrataan sitä tietokantaan tallennettuun salattuun salasanaan. Jos salasanat eivät täsmää niin lähetetään palvelimelle virheviesti. Seuraavassa vaiheessa palvelin luo käyttäjälle tokeni, jonka avulla todennetaan käyttäjä kirjautuneeksi sovelluksessa. Viimeisessä vaiheessa lisätään tokeni JSON- olion sisälle ja palautetaan JSON- data sovellukseen. Tokenilla tarkoitetaan tässä tapauksessa Access Tokenia, joka pitää sisällään oikeudet käyttää suojattua sisältöä Azuressa.



KUVA 8. Kirjautumisen vuokaavio

3.2 Sisällön lataaminen

Kirjautumisen jälkeen sovellus lataa kaiken sisällön paikalliseen tietokantaan talteen, jotta sovelluksen käytön aikana ei tarvitse tehdä turhia palvelin pyyntöjä. Enne sisällön lataamista käyttäjä pitää olla kirjautuneena. Sisällön lataamisen rajapinta vaatii sisäänkirjautumisen. Jos käyttäjä ei ole kirjautunut sisään niin palvelin palauttaa virhe ilmoituksen. Tarvittavia tietoja ovat kaikki käyttäjälle tarkoitetut ryhmät, tehtävät ja materiaalit. Lataus näytössä ei näy muuta, kuin Puhti logo ja pieni lataus animaatio. (Kuva 9.) Lataus näytön käynnistyessä sovellus lähettää palvelimelle pyynnön. Palvelin pyynnön mukana menee tieto käyttäjästä, jonka perusteella ladataan tarvittava tieto sovellukseen.



KUVA 9. Sisällön lataaminen

Kun pyyntö on saapunut palvelimelle niin ensin ladataan kaikki käyttäjälle kuuluvat ryhmät tietokannasta. Tämän jälkeen ladataan ryhmille kuuluvat materiaalit ja materiaaleille kuuluvat tehtävät. Viimeisenä ladataan kaikki tehtävät, jotka ei liity materiaaliin, mutta liittyy ryhmään. Lopuksi palautetaan data sovellukseen. Kun pyyntö on saapunut sovellukseen niin sovellus tallentaa pyynnön datan paikalliseen tietokantaan. Tämän jälkeen kaiken tiedon voi hakea helposti paikallisesta muistista, eikä tarvitse tehdä koko ajan palvelin pyyntöjä tietokantaan.

Koska tehtäviin ja materiaaleihin voi liittyä tiedostoja niin tiedostot pitää ladata erikseen Azuren tiedosto varastosta eli Azure Storagesta. Sisällön latauksen yhteydessä tulee SAS- osoite, joka antaa oikeuden ladata tiedostoja Azuresta.

Kun kaikki osoitteet on muodostettu niin sovellus lataa kaikki tiedostot laitteen muistiin. Sovellus tarkistaa aina, että onko tiedosto jo aiemmin ladattu sovelluksen sisälle. Jos tiedosto on olemassa niin ei ladata samaa tiedostoa turhaan uudestaan. Tämä vähentää huomattavasti latausaikaa ensimmäisen kirjautumisen jälkeen. Lopuksi sovellus avaa etusivun käyttäjälle, jossa näkyy ladatut tiedot valmiina.

3.3 Etusivu

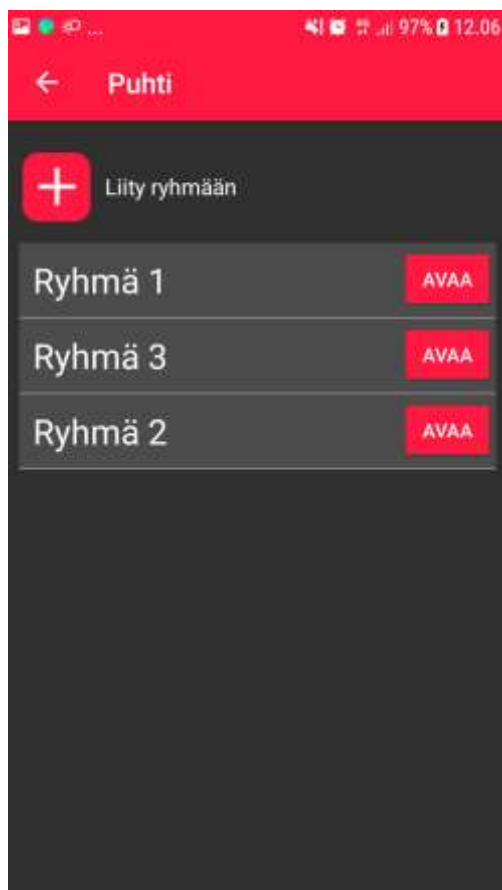
Kuvassa 10 näkyy etusivun näyttö. Etusivulla näkyy kaikki aktiivisena oleva materiaali. Materiaalit ovat karusellinäytössä, joka vaihtaa materiaalia automaattisesti 5 sekunnin välein. Etusivua voi kuvitella esittelynäyttönä, jossa esitellään saatavilla olevaa materiaalia. Materiaalia voi myös selata vetämällä karusellinäyttöä vasemmalle tai oikealle. Yhdessä materiaalikortissa näkyy materiaalin nimi, kuva ja kuvaus. Materiaalit muodostuvat PDF-tiedostoista, jotka ladataan heti kirjautumisen jälkeen. PDF-tiedostot tallennetaan laitteen paikalliseen muistiin, josta ne saa ladattua sovellukseen. Etusivulla näkyvän materiaalikortin kuva muodostuu PDF-tiedoston ensimmäisestä sivusta. Materiaalin kuvaa painamalla pääsee selaamaan PDF-tiedostoa ja tekemään materiaaliin liittyviä tehtäviä.



KUVA 10. Etusivu

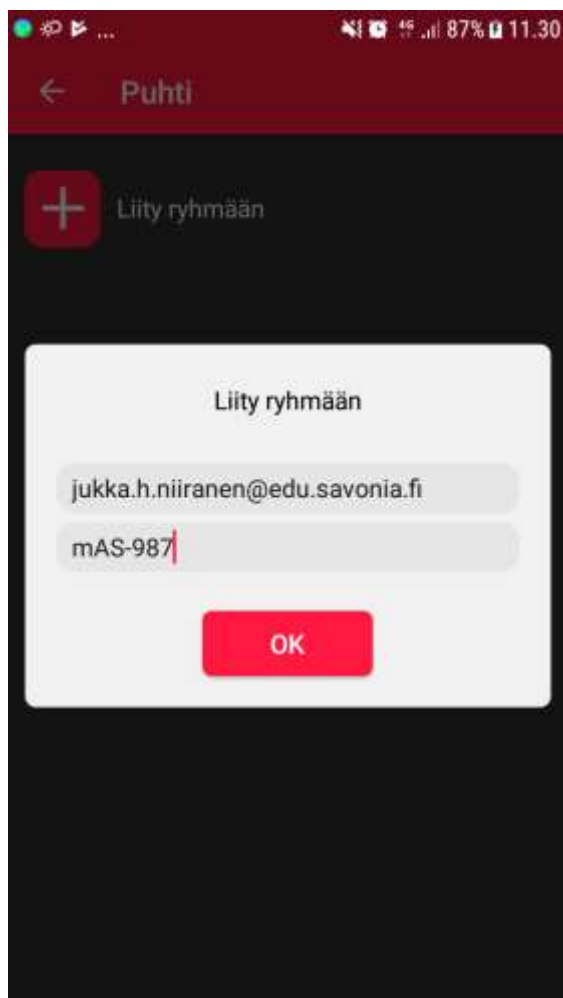
3.4 Ryhmät

Kaikki sovelluksessa näytettävät materiaalit ja tehtävät liittyvät eri ryhmiin. Etusivun yläpalkissa näkyy nykyisen ryhmän nimi ja sitä painamalla avautuu lista ryhmistä, joihin käyttäjä kuuluu. (Kuva 11)



KUVA 11. Ryhmälista

Ennen kuin sovelluksessa voidaan näyttää mitään materiaalia niin käyttäjän täytyy liittyä johonkin ryhmään. Ryhmälistan yläpuolella näkyy punainen plusnappi, jolla käyttäjä voi liittyä uuteen ryhmään. Nappia painamalla avautuu dialogi johon pitää syöttää valmentajan sähköpostiosoite ja ryhmäkoodi, jonka valmentaja on lähettänyt asiakkaalle. (Kuva 12)



KUVA 12. Ryhmään liittymisen dialogi

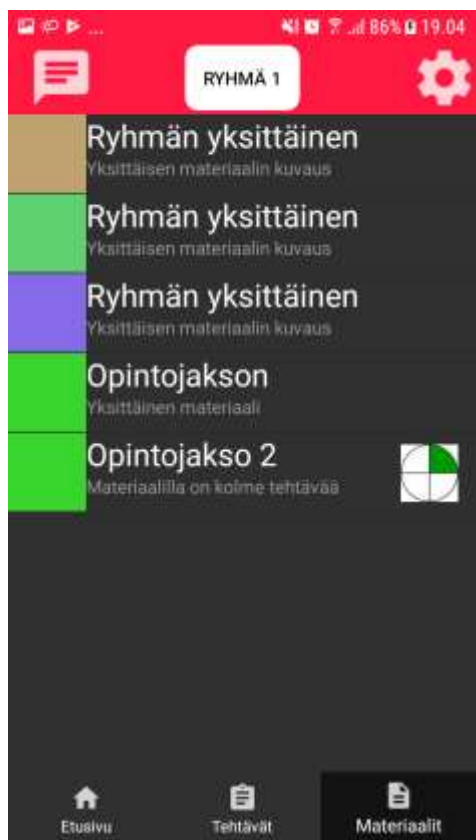
Kun painaa "ok" nappia niin sovellus lähettää palvelimelle kutsun, jossa käyttäjä liitetään haluttuun ryhmään. Jos ryhmään liittyminen palvelimen puolella on onnistunut niin sovellus ilmoittaa käyttäjälle onnistuneesta ryhmään liittymisestä ja sovellus avaa lataus näytön uudestaan. (Kuva 9) Lataus näyttö synkronoi sovelluksen tiedot palvelimen kanssa uudestaan, jotta saadaan uuden ryhmän tiedot sovellukseen.

Kun käyttäjä kuuluu johonkin ryhmään tai ryhmiin niin ryhmälistasta voi valita ryhmän. Painamalla valitse nappia ryhmän kohdalta sovellus avaa etusivun ja lataa kyseisen ryhmän materiaalit näkyviin. Aina kun sovellus avataan uudestaan niin sovellus muistaa, että mikä ryhmä oli viimeksi päällä ja avaa sen sisällön.

3.5 Materiaalit

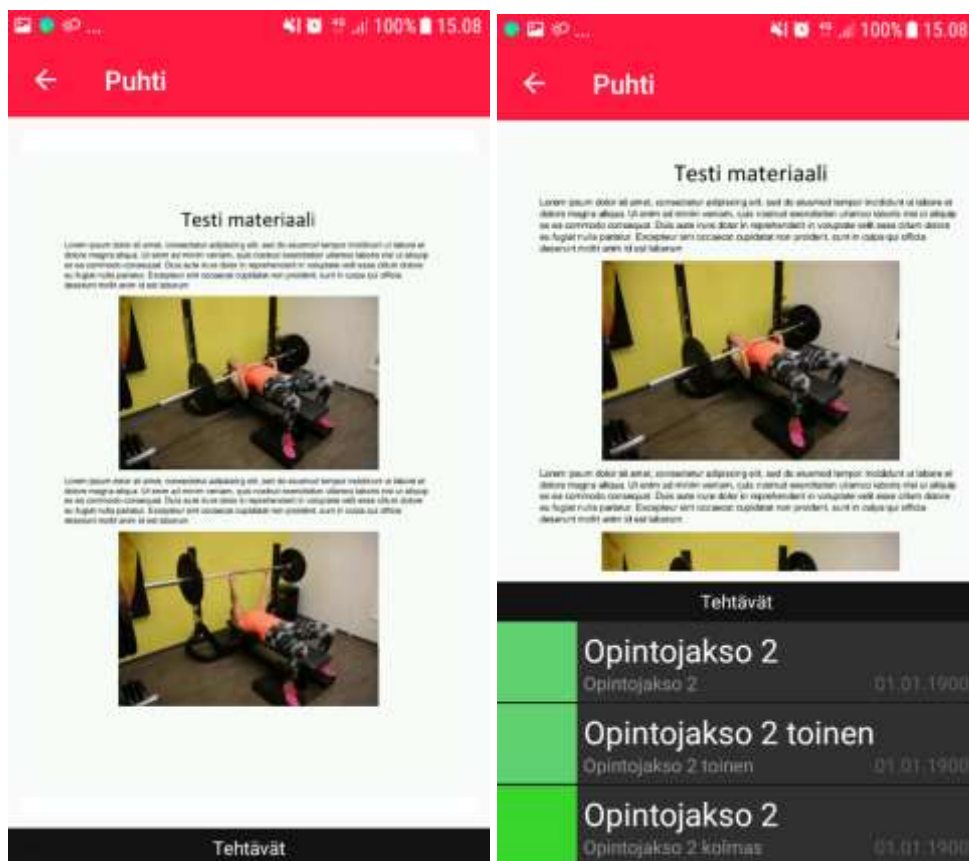
Saatavilla olevaa materiaalia voi selata etusivun lisäksi materiaalit välilehdestä. Kuvassa 13 olevassa listassa on listattu kaikki aktivoituneet materiaalit riippumatta siitä, että onko niitä suoritettu vai ei. Materiaalit tulevat aktiiviseksi, jos niiden aloitus päivämäärä on alkanut. Yhden materiaalin oikealla puolella näkyy piirakkaikoni, joka vastaa miten paljon materiaalin tehtävistä on suoritettu. Piirakkakuvio ei ole dynaaminen vain se näyttää suoritukset neljäsosan tarkkuudella. Listan vasemmalla puolella näkyy materiaaleilla omat tunnistevärit, jotka valmentaja luo valmentajan

sovelluksessa. Listan materiaalia painamalla avautuu sivu, jossa näkyy kyseisen materiaalin PDF-tiedosto.



KUVA 13. Materiaalit lista

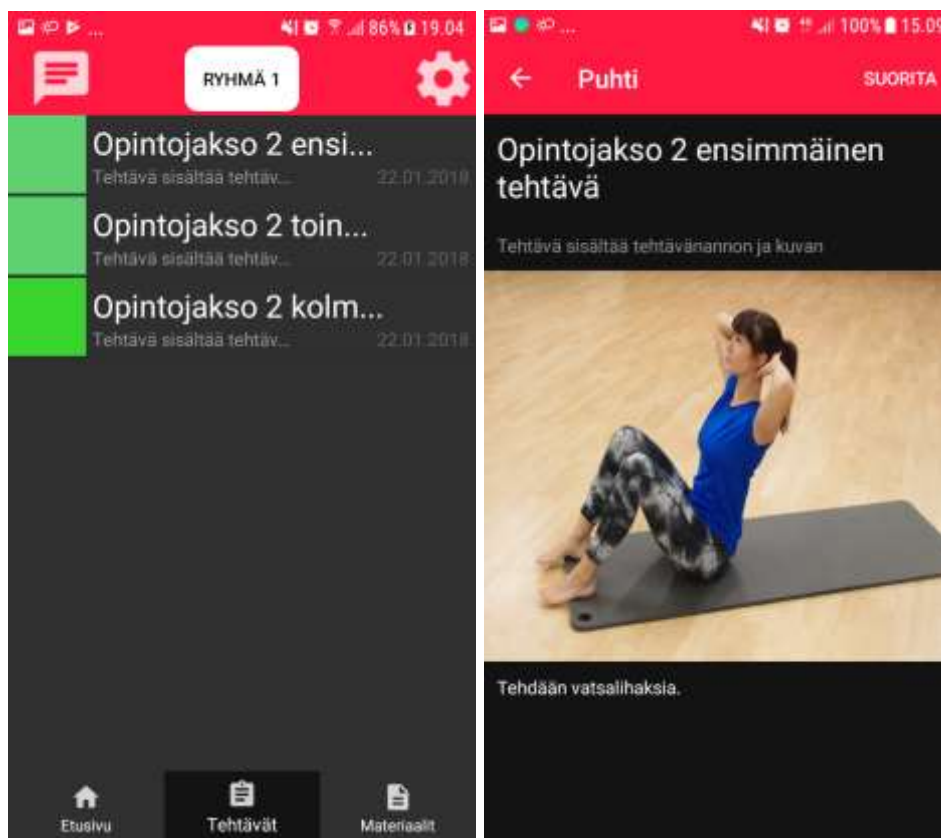
Kuvassa 14 näkyy PDF-tiedosto materiaali sivulla. PDF-näkymän alapuolella näkyy lista materiaaliin kuuluvista tehtävistä. Listan voi laajentaa tai pienentää painamalla tehtävät otsikkoa. Tehtävää painamalla avautuu tehtävänäkymä.



KUVA 14. Materiaalinselausnäkymä

3.6 Tehtävät

Kuvassa 15 näkyy tehtävät välilehden lista ja tehtävän tarkastelunäkymä. Listassa on kaikki ryhmälle ja ryhmän materiaaleille kuuluvat tehtävät. Tehtävää painamalla avautuu tehtävänäkymä, jossa näkyy valmentajan luoma tehtävä.



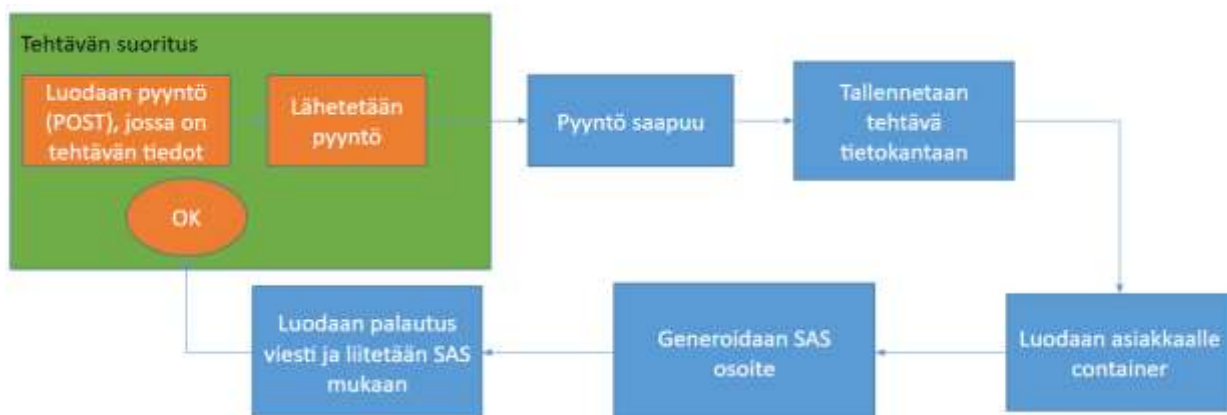
KUVA 15. Tehtävät lista ja tehtävänäkymä

Tehtävänäkymässä näkyy tehtävän nimi, kuvaus, mediatiedosto ja tehtävän selitys. Kun tehtävänäkymä avautuu niin sovellus lataa tehtävän tiedot paikallisesta tietokannasta ja hakee tehtävän medialle SAS-osoitteen, jotta sovellus voi ladata mediatiedoston sovellukseen. Media tiedosto voi olla JPG, MP3 tai MP4 tyyppinen tiedosto. Tehtävä näkymän komponentit muodostuvat sen mukaan minkälaisen tehtävän valmentaja on asettanut asiakkaalle. Esimerkiksi valmentaja luo tehtävän video muodossa niin kuva ja äänensoittimen komponentit ovat piilossa ja videon komponentti on näkyvillä. Sivun oikeassa yläkulmassa näkyy suorita nappi, jolla voi aloittaa tehtävän teon.



KUVA 16. Tehtäväntekonäkymä

Kuvassa 16 näkyy tehtäväntekonäkymä, missä käyttäjä voi suorittaa tehtävän. Suoritettavan tehtävän tyyppi riippuu siitä mitä valmentaja haluaa asiakkaalta. Valmentaja voi valmentajan sovelluksesta määrätä tietyille tehtävälle tehtävän tyyppin, joka voi olla video, kuva tai ääni tyyppinen tehtävä. Tallenna napin yläpuolella näkyy kolme hymiota joiden avulla käyttäjä voi ilmaista, että oliko tyytyväinen annettuun tehtävään. Kun käyttäjä on täyttänyt kaikki vaaditut tehtävän osat niin tallenna napista sovellus lähettää tiedot palvelimelle ja tallentaa suoritettavan tehtävän paikalliseen tietokantaan. Kuvassa 19 näkyy tehtävän suorituksen vuokaavio. Ensimmäisen lisätään kaikki suorituksen tiedot tietokantaan. seuraavaksi luodaan asiakkaalle containeri palvelimelle mediaa varten, jos sitä ei ole vielä olemassa. Tämän jälkeen luodaan SAS osoite median lähetystä varten ja palautetaan tiedot sovellukseen.



KUVA 19. Suoritettavan tehtävän vuokaavio

Kun palvelin on palauttanut SAS- osoitteen niin lähetetään tehtävän tiedosto palvelimelle. Tämän jälkeen tehtävän lähetys ja tallentaminen on valmis, joten sovellus ohjaa käyttäjän takaisin etusivulle.

3.7 Viestit

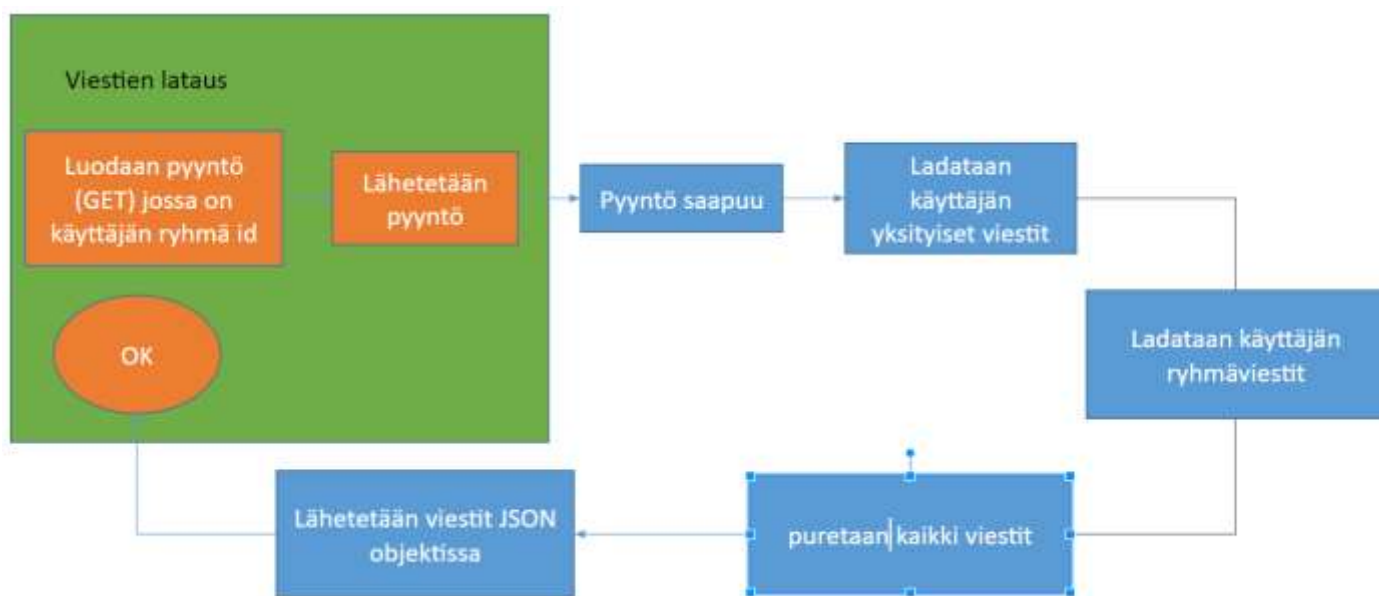
Viestejä pääsee tarkastelemaan, kun painaa etusivun vasemmassa yläkulmassa olevaa viesti-ikonin. Viesti-ikonista avautuu lista, jossa näkyy kaikki keskustelut. (Kuva 18) Keskustelulistassa näkyy kaikki ryhmät ja ryhmien valmentajat erikseen. Eli viestejä voi lähettää, joko ryhmälle tai ryhmän valmentajalle. Yhtä keskustelua painamalla avautuu keskustelunäkymä.



KUVA 18. Keskustelulista

Keskustelunäkymässä näkyy kaikki kyseisen keskustelun viestit. (Kuva 20) Ennen kuin viestit näytetään näytössä niin ne ladataan palvelimelta laitteen paikalliseen muistiin. Viestitys tässä näkymässä ei ole reaaliaikaista vaan ne pitää ladata aina palvelimelta. Kun sovellus lähettää pyynnön palvelimelle niin sovellus lähettää tiedon käyttäjästä ja ryhmästä, jonka perusteella haetaan keskustelun viestit.

Kuvassa 19 näkyy viestien latauksen vuokaavio. Ensin ladataan kaikki yksityisviestit ja ryhmäkeskustelut. Kaikki viestit ovat salattuja tietokannassa, joten ne pitää purkaa ennen kuin niitä voi lähettää sovellukseen. Viimeisessä vaiheessa palvelin lähettää viestit sovellukseen.



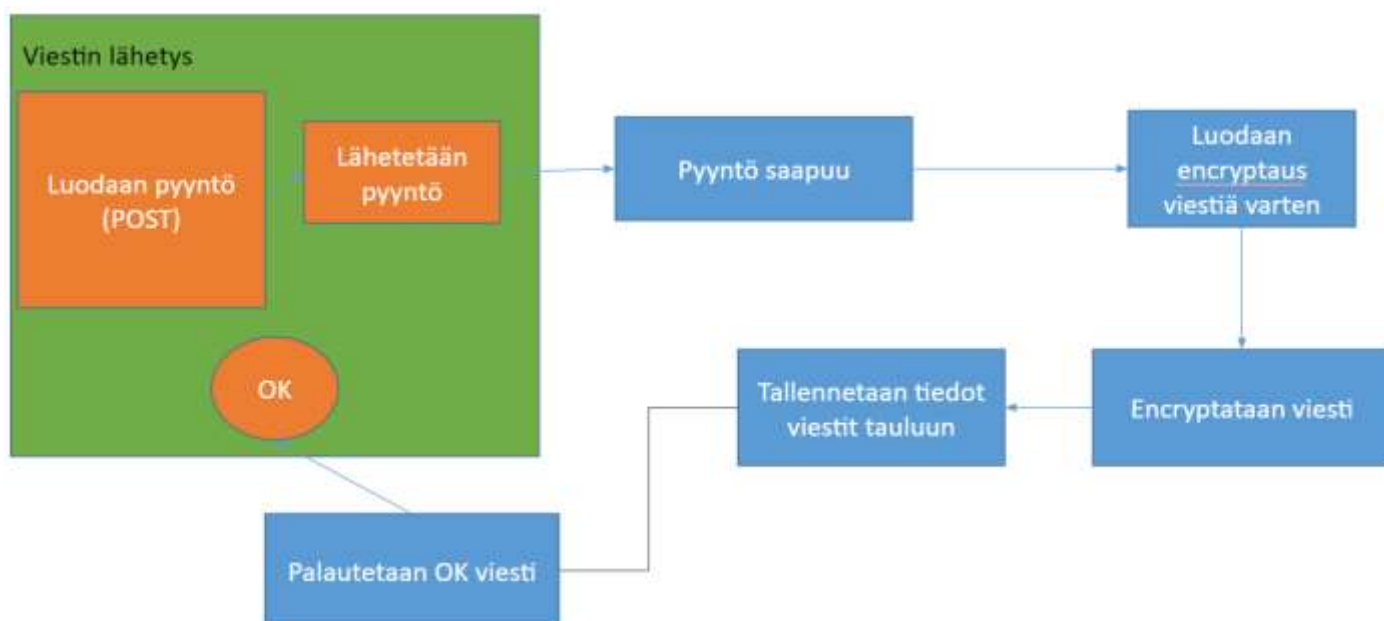
KUVA 19. Viestien latauksen vuokaavio

Viestinäköymän kaikki omat viestit näkyvät näytän oikealla puolella ja muut viestit näkyvät vasemmalla puolella. Uuden viestin voi lähettää kirjoittamalla viestin näytön alapuolella olevaan kirjoitus kenttään ja painamalla lähetä nappia. Lähetä napista sovellus tallentaa viestin paikalliseen tietokantaan ja lähettää palvelimelle tiedon käyttäjästä, keskustelusta ja viestistä.



KUVA 20. Keskustelunäkymä

Kuvassa 21 näkyy viestin lähetyksen vuokaavio. Kun pyyntö saapuu niin palvelin salaa lähetetyn viestin. Tämän jälkeen viesti tallennetaan tietokantaan ja palvelin lähettää ok viestin sovellukseen. Vuokaaviossa ei käsitellä kuin yksi viesti kerrallaan.



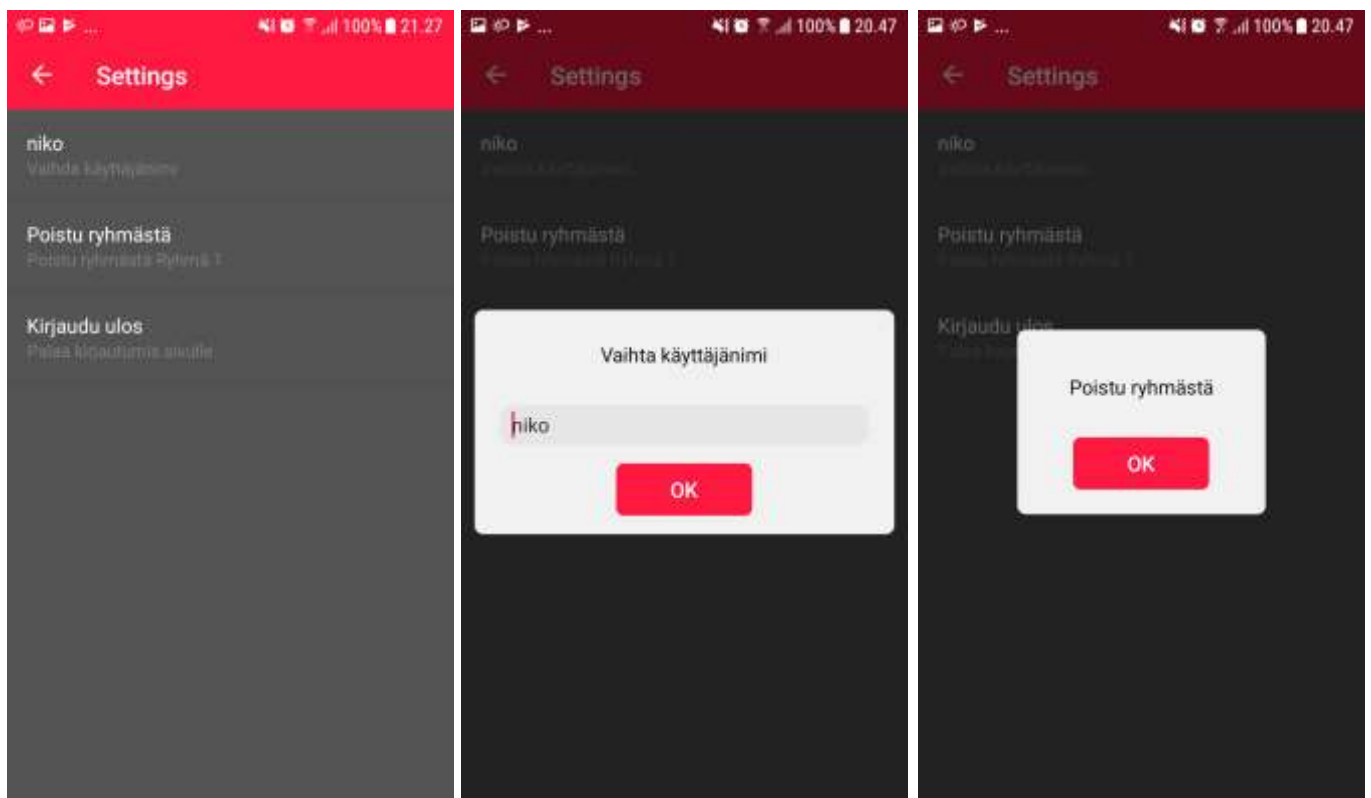
Kuva 21. Viestin lähetyksen vuokaavio

3.8 Asetukset

Asetukset sivulle (Kuva 22) pääsee painamalla etusivun oikeassa yläkulmassa olevaa asetukset nappia. Asetukset sivulla voi vaihtaa käyttäjänimen, poistua ryhmästä tai kirjautua ulos sovelluksesta. Painamalla käyttäjänimeä avautuu dialogi, johon voi syöttää uuden käyttäjänimen. Kun painaa ok nappia niin sovellus lähettää palvelimelle pyynnön, jonka mukana menee tieto käyttäjästä ja uudesta nimimerkistä.

Jos käyttäjä haluaa poistua nykyisestä ryhmästä niin painamalla poistu ryhmästä kohtaa avautuu dialogi, jossa voi vahvistaa poiston. Käyttäjä poistuu ryhmästä mikä on tällä hetkellä päällä sovelluksessa. Sovellus lähettää palvelimelle pyynnön minkä mukana menee tieto käyttäjästä ja nykyisestä ryhmästä. Kun palvelimelta tulee vastaus niin sovellus lataa ja päivittää tiedot uudestaan sovellukseen. Sovellus poistaa ryhmän viestinnän ja sille kohdennetun sisällön laitteen muistista.

Kirjaudu ulos valinnasta sovellus ensin kirjaa käyttäjän ulos palvelimelta. Tämän jälkeen käyttäjä ei voi käyttää rajapintoja, jotka vaativat kirjautumisen. Sovellus tämän jälkeen ohjaa käyttäjän kirjautumis-sivulle.



KUVA 22. Asetukset

4 YHTEENVETO

Lopputuloksena tästä opinnäytetyöstä tuli toimiva sovellus. Tämän sovelluksen tekeminen venyi aika pitkälle, koska opinnäytetyön aihe oli todella laaja. Ulkoasu sovelluksessa ei vielä näytä kaikissa sovelluksen sivuissa hyvälle. Tähän projektiin ei saatu graafista suunnittelijaa, joten kaikki ulkoasu muokkaukset on suunniteltu itse. Ongelmia eniten tuotti tiedostojen lähettäminen ja etusivun karusellinäyttö. Ongelmia myös tuotti rajapintojen tekeminen yhteensopivaksi tämän sovelluksen ja valmentajan sovelluksen välillä. Esimerkiksi materiaalien ja tehtävien lataamisen rajapinnan koodista tuli paljon monimutkaisempi, mitä olisi tarvinnut. Rajapinnan olisi voinut suunnitella huolellisemmin etukäteen. Itse olin kuitenkin tyytyväinen lopputulokseen. Opinnäytetyön aikana opin paljon uutta Android- ohjelmoinnista ja Microsoft Azuren palveluista.

Sovellus on nyt toteutettu asiakkaan toiveiden mukaan, mutta mielestäni sovelluksesta puuttuu vielä muutama ominaisuus mitä olisi hyvä olla. Sovelluksessa ei tällä hetkellä näy missään, että mitkä tehtävät on suoritettu ja mitkä ei. Kirjautumissivulle myös voisi tehdä salasanan nollaus ominaisuuden, jos käyttäjä unohtaa salasanan. Sovelluksen viestintä ominaisuudesta voisi myös tehdä reaaliaikaisen.

LÄHTEET JA VIITTAUKSET

ANDROID DEVELOPERS. Android Studio. [Viitattu 2018-05-03.] Saatavissa:

<https://developer.android.com/studio/index.html>

ANDROID DEVELOPERS 2018. Build a UI with Layout Editor. [Viitattu 2018-05-03.] Saatavissa:

<https://developer.android.com/studio/write/layout-editor.html>

ANDROID DEVELOPERS 2018. Run Apps on the Android Emulator. [Viitattu 2018-05-05.] Saatavissa:

<https://developer.android.com/studio/run/emulator.html>

ANDROID DEVELOPERS 2018. Configure Your Build. [Viitattu 2018-05-05.] Saatavissa:

<https://developer.android.com/studio/build/>

MICROSOFT AZURE 2018. What is Azure? [Viitattu 2018-05-06.] Saatavissa:

<https://azure.microsoft.com/en-gb/overview/what-is-azure/>

MICROSOFT AZURE 2017. Web Apps overview. [Viitattu 2018-05-06.] Saatavissa:

<https://docs.microsoft.com/en-us/azure/app-service/app-service-web-overview>

MICROSOFT AZURE 2018. Introduction to Azure Storage. [Viitattu 2018-05-06.] Saatavissa:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

MICROSOFT AZURE 2017. Using shared access signatures (SAS). [Viitattu 2018-05-06.] Saatavissa:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-dotnet-shared-access-signature-part-1>